

# Practica 2: Análisis descriptivo e inferencial

*José Luis Melo*

*Lisardo Gayán Tremps*

*1 June, 2019*

## Contents

<b>1 - Descripción del dataset. ¿Por qué es importante y qué pregunta/problema pretende responder?</b>	<b>1</b>
<b>2 - Integración y selección de los datos de interés a analizar</b>	<b>2</b>
<b>3 - Limpieza de datos</b>	<b>4</b>
3.1. ¿Los datos contienen ceros o elementos vacíos? ¿Cómo gestionaría cada uno de estos casos? .	4
3.2. Identificación y tratamiento de valores extremos.** . . . . .	6

```
library(kableExtra)
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:kableExtra':
##
##      group_rows

## The following objects are masked from 'package:stats':
##
##      filter, lag

## The following objects are masked from 'package:base':
##
##      intersect, setdiff, setequal, union
```

## 1 - Descripción del dataset. ¿Por qué es importante y qué pregunta/problema pretende responder?

El dataset de Titanic: Machine Learning from Disaster se registran los datos de los pasajeros del famoso trasatlántico y se utiliza para predecir los sobrevivientes. Los datos están divididos en dos datasets, uno de test y otro entrenamiento, para la creación de modelos de predicción.

## 2 - Integración y selección de los datos de interés a analizar

Siendo los dataset relativamente pequeños (dataset de test 418 observaciones y 11 variables, dataset de train 891 y 12 variables), con solo 1309 observaciones y 11 variables, se importaran en su totalidad para la realización de esta practica. Inicialmente no se importara la variable survived del dataset de train. A continuacion se importan, fusionan los datos y se revisa la estructura inicial de los datos.

```
datostest <- read.csv("./data/test.csv", stringsAsFactors = F)
datostrain <- read.csv("./data/train.csv", stringsAsFactors = F)
datostrain1 <- datostrain[,-2]
```

```
data <- rbind(datostrain1, datostest) # Fusion datasets
str(data)
```

```
## 'data.frame': 1309 obs. of 11 variables:
## $ PassengerId: int 1 2 3 4 5 6 7 8 9 10 ...
## $ Pclass : int 3 1 3 1 3 3 1 3 3 2 ...
## $ Name : chr "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
## $ Sex : chr "male" "female" "female" "female" ...
## $ Age : num 22 38 26 35 35 NA 54 2 27 14 ...
## $ SibSp : int 1 1 0 1 0 0 0 3 0 1 ...
## $ Parch : int 0 0 0 0 0 0 0 1 2 0 ...
## $ Ticket : chr "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
## $ Fare : num 7.25 71.28 7.92 53.1 8.05 ...
## $ Cabin : chr "" "C85" "" "C123" ...
## $ Embarked : chr "S" "C" "S" "S" ...
```

A continuación se detallan las variables y su tipo inicial, este ultimo, se modificara para su mejor analisis.

```
tipos <- sapply(data, class)
kable(data.frame(Variables = names(tipos), Tipo_Variable= as.vector(tipos))) %>%
  kable_styling(bootstrap_options = "striped", full_width = F, position = "left")
```

Variables	Tipo_Variable
PassengerId	integer
Pclass	integer
Name	character
Sex	character
Age	numeric
SibSp	integer
Parch	integer
Ticket	character
Fare	numeric
Cabin	character
Embarked	character

De las variables, se convertiran a entero la edad y a factor las siguientes variables Pclass, Sex, Cabin y Embarked.

```
data$Age <- as.integer(data$Age)
data$Pclass <- as.factor(data$Pclass)
data$Sex <- as.factor(data$Sex)
data$Embarked <- as.factor(data$Embarked)
```

```
data$Cabin <- as.factor(data$Cabin)

tipos_new <- sapply(data, class)
kable(data.frame(Variables = names(tipos_new), Tipo_Variable= as.vector(tipos_new))) %>%
  kable_styling(bootstrap_options = "striped", full_width = F, position = "left")
```

Variables	Tipo_Variable
PassengerId	integer
Pclass	factor
Name	character
Sex	factor
Age	integer
SibSp	integer
Parch	integer
Ticket	character
Fare	numeric
Cabin	factor
Embarked	factor

Una vez modificadas los tipos de valores se resume que

```
summary(data)
```

```
##   PassengerId   Pclass      Name      Sex      Age
##   Min.      :    1   1:323   Length:1309   female:466   Min.      : 0.00
##   1st Qu.:   328   2:277   Class :character   male  :843   1st Qu.:21.00
##   Median :   655   3:709   Mode  :character           Median :28.00
##   Mean    :   655                                Mean    :29.86
##   3rd Qu.:   982                                3rd Qu.:39.00
##   Max.    :  1309                                Max.    :80.00
##                                           NA's    :263
##      SibSp      Parch      Ticket      Fare
##   Min.      :0.0000   Min.      :0.0000   Length:1309   Min.      : 0.000
##   1st Qu.:0.0000   1st Qu.:0.0000   Class :character   1st Qu.: 7.896
##   Median :0.0000   Median :0.0000   Mode  :character   Median : 14.454
##   Mean    :0.4989   Mean    :0.385    Mean    : 33.295
##   3rd Qu.:1.0000   3rd Qu.:0.000    3rd Qu.: 31.275
##   Max.    :8.0000   Max.      :9.000    Max.    :512.329
##                                           NA's     :1
##      Cabin      Embarked
##           :1014      : 2
##   C23 C25 C27      : 6   C:270
##   B57 B59 B63 B66: 5    Q:123
##   G6              : 5    S:914
##   B96 B98          : 4
##   C22 C26          : 4
##   (Other)         : 271
```

PassengerId: Variable de tipo entero que contiene el id del pasajero, no existen valores nulos o perdidos.

Pclass: Variable de tipo factor con la categoria asignada al pasajero, no existen valores nulos o perdidos.

Name: Variable de tipo texto con el nombre del pasajero, no existen valores nulos o perdidos.

Sex: Variable de tipo factor con el genero del pasajero, no existen valores nulos o perdidos.

Age: Variable de tipo entero que especifica la edad del pasajero, **existen 263 valores nulos**.

SibSp: Variable de tipo entero que especifica el numero de hermanos/esposa abordo, no existen valores nulos o perdidos.

Parch: Variable de tipo entero que especifica el numero de padres/hijos abordo, no existen valores nulos o perdidos.

Ticket: Variable de tipo texto que indica el numero de ticket, no existen valores nulos o perdidos.

Fare: Variable de tipo numero que especifica la tarifa pagada, **existe 1 valor nulo**.

Cabin: Variable de tipo factor donde se especifica la cabina asignada, **existen 1014 valores perdidos**.

Embarked: Variable de tipo factor que indica el puerto de embarque, **existen 2 valores perdidos**.

## 3 - Limpieza de datos

### 3.1. ¿Los datos contienen ceros o elementos vacíos? ¿Cómo gestionarías cada uno de estos casos?

De las variables existentes a continuación se especifican aquellas que contienen valores perdido o nulos.

Age: *existen 263 valores nulos*.

“Para imputar valores **edad**, (analizando si es mejor imputar mediante la media de los valores, rpart o mice)”

<http://jstatsoft.org/article/view/v045i03>

Data Analysis with R, A comprehensive guide to manipulating, analyzing and visualizing data in R Pag 373 - 386

```
sum(is.na(data$Age))      ## Realiza el conteo de valores NA de la variable Age
```

```
## [1] 263
```

Fare: *existe 1 valor nulo*.

Para imputar valores **Fare**

Dado que unicamente hay un valor perdido, es posible imputarlo por la media en base al puerto de embarque “S” y la clase “3”

```
sum(is.na(data$Fare))      ## Realiza el conteo de valores NA de la variable Fare
```

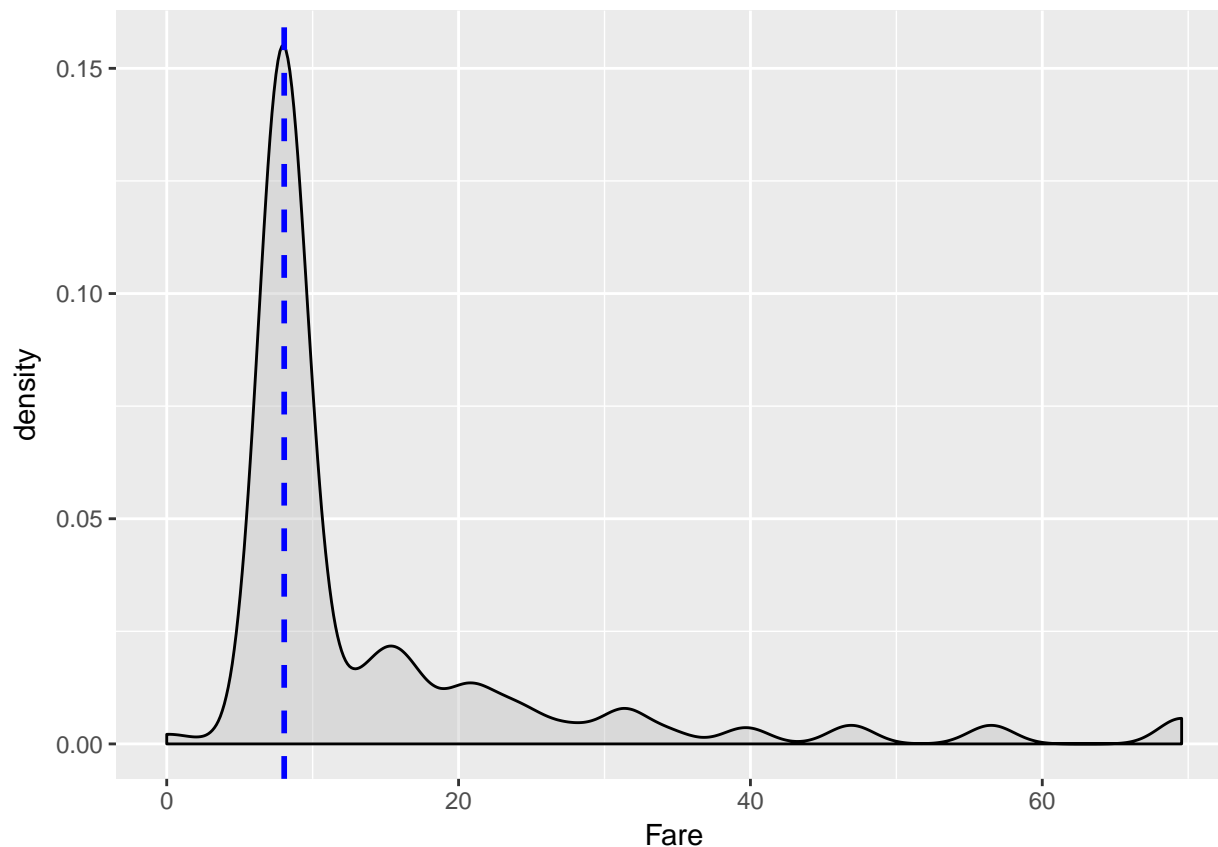
```
## [1] 1
```

```
M_fare<- subset(data,data$Pclass == '3' & data$Embarked == 'S')
median(M_fare$Fare, na.rm = T)
```

```
## [1] 8.05
```

```
ggplot(M_fare, aes(x = Fare)) +
  geom_density(fill = 'grey', alpha=0.4) +
  geom_vline(aes(xintercept=median(Fare, na.rm=T)),
    colour='blue', linetype='dashed', lwd=1)
```

```
## Warning: Removed 1 rows containing non-finite values (stat_density).
```



La tarifa de 80 coincide con la media de los pasajeros de primera clase que embarcaron en C, por lo que se podría imputar este puerto.

```
data$Fare[c(1044)] <- 80
```

Cabin: *existen 1014 valores perdidos.*

Para imputar valores **Cabin**

Esta variable tiene muchos valores perdidos, se podría conseguir predecir la cubierta asignada al pasajero pero es un dato que poco beneficio podría traer ya que se puede realizar el analisis con la combinación entre la tarifa y la clase del pasajero.

```
sum(data$Cabin=="")      ## Realiza el conteo de valores vacios de la variable Cabin
```

```
## [1] 1014
```

Embarked: *existen 2 valores perdidos.*

Para imputar valores **Embarked**

Al ser unicamente dos valores perdidos, se podría sustituir los valores por la media, en base a otros pasajeros de la misma clase y puerto de embarque. Los pasajeros han pagado una tarifa de 80 y pertenecian a primera clase.

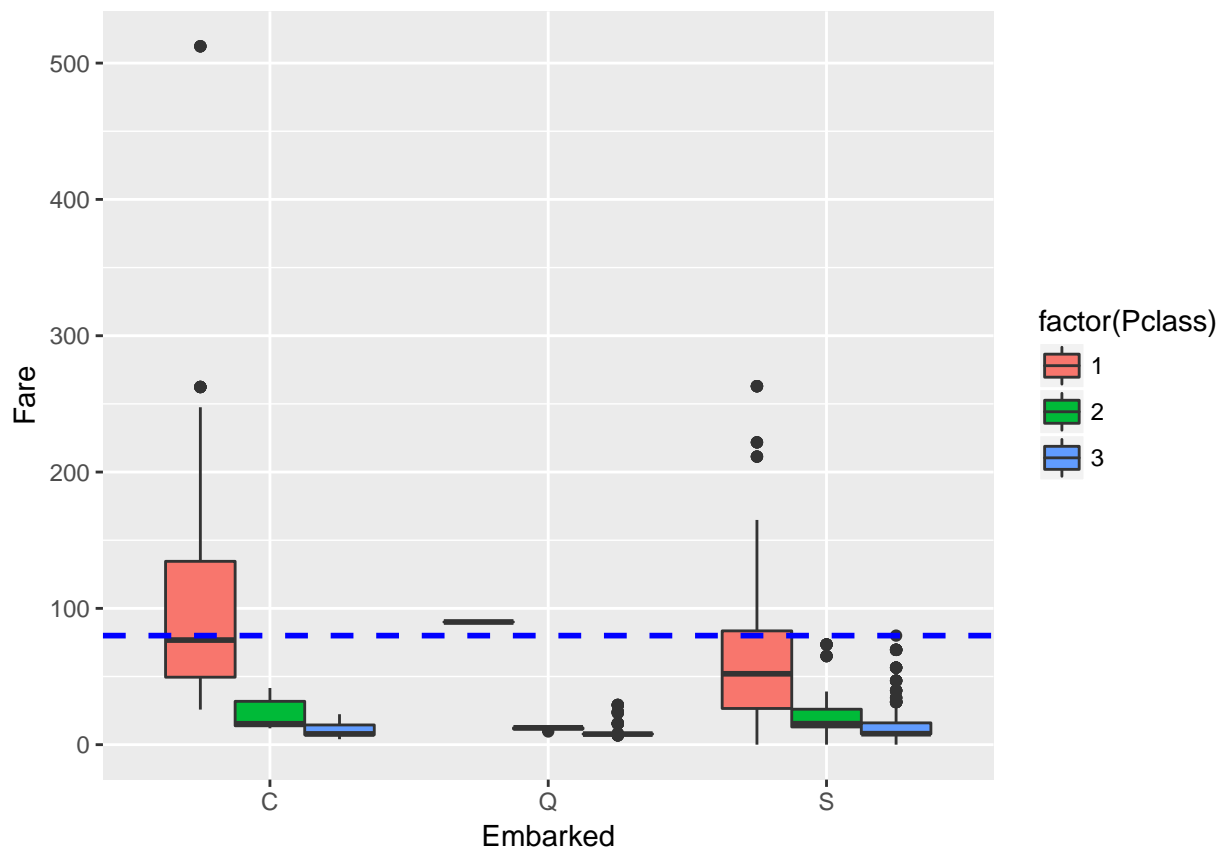
```
data[(data$Embarked==""),]
```

```
##      PassengerId Pclass      Name      Sex
## 62             62      1  Icard, Miss. Amelie female
```

```
## 830      830      1 Stone, Mrs. George Nelson (Martha Evelyn) female
##      Age SibSp Parch Ticket Fare Cabin Embarked
## 62   38     0     0 113572   80   B28
## 830  62     0     0 113572   80   B28
```

```
embarco <- data %>%
  filter( PassengerId != 62 & PassengerId != 830)

ggplot(embarco, aes(x = Embarked, y = Fare, fill = factor(Pclass))) +
  geom_boxplot() +
  geom_hline(aes(yintercept=80),
    colour='blue', linetype='dashed', lwd=1)
```



La tarifa de 80 coincide con la media de los pasajeros de primera clase que embarcaron en C, por lo que se podría imputar este puerto.

```
data$Embarked[c(62, 830)] <- 'C'
```

### 3.2. Identificación y tratamiento de valores extremos.\*\*

```
##par(mfrow=c(1,3))
##age.bp <- boxplot(data$Age, main= "Edad")
##hist (data$Age, main = "Dist. Edad")
```